



# Création Semi-Automatique d'un Modèle Numérique de Terrain

Joachim Pouderoux, Jean-Christophe Gonzato, Pascal Guitton

## ► To cite this version:

Joachim Pouderoux, Jean-Christophe Gonzato, Pascal Guitton. Création Semi-Automatique d'un Modèle Numérique de Terrain. Actes des 16èmes Journées de l'AFIG, 2003, France. pp.151-160. hal-00308006

**HAL Id: hal-00308006**

**<https://hal.science/hal-00308006>**

Submitted on 20 Jan 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Création Semi-Automatique d'un Modèle Numérique de Terrain

Joachim Pouderoux, Jean-Christophe Gonzato, Pascal Guitton

INRIA Futurs, LaBRI - 351, cours de la Libération - 33400 Talence  
{pouderou | gonzato | guitton}@labri.fr

**Résumé :** *Un modèle numérique de terrain permet de modéliser une zone géographique et principalement son relief. Les applications pratique de ces modèles sont innombrables. Dans cet article, nous décrivons un processus permettant de créer un modèle numérique de terrain à partir d'une carte topographique couleur numérisée. Nous présentons d'abord des techniques classiques d'extraction des courbes de niveaux. Un algorithme de reconnexion automatique des courbes altérées est ensuite proposé. Enfin, nous présentons différentes techniques d'interpolation de terrain et explorons notamment une interpolation par fonctions de bases radiales et une technique de partitionnement de l'unité.*

**Mots-clés :** MNT, analyse d'image, extraction d'information, traitement des courbes de niveaux, interpolation, RBF

## 1 Introduction

Un modèle numérique de terrain (MNT) est une représentation informatique d'une zone géographique et de son relief. Un MNT élémentaire peut se réduire à un tableau de nombres représentant la distribution spatiale des altitudes d'un terrain. On l'appelle alors aussi modèle numérique d'élévation (MNE ou *Digital Elevation Terrain, DEM*). Les débouchés pratiques de ces modèles sont innombrables. Ils sont notamment utilisés en réalité virtuelle, en géologie, en ingénierie civile et militaire, en hydrographie [MGL91], dans les systèmes d'informations géographiques (SIG), en géomorphologie, etc.

**Problématique.** De grands instituts tels que l'Institut Géographique National (IGN, en France) ou l'U.S. Geological Survey (USGS, aux Etats-Unis) proposent de larges bases de données de modèles numériques de terrains à différentes échelles et indexant de nombreuses informations. Cependant ces données sont très onéreuses. Cette solution n'est donc pas envisageable dans le cadre d'applications de recherche ou grand public. De plus, une quantité innombrable de cartes topographiques dessinées manuellement avant l'émergence de l'informatique ne sont disponibles que dans leur version originale papier, et une version numérique pourrait être très intéressante pour des historiens.

La création de modèles numériques de terrain à partir de cartes topographiques papier numérisées est une technique à moindre coût. Le principe repose sur un traitement de l'image numérique. Il s'agit d'en extraire les informations nécessaires et particulièrement les courbes de niveaux. Une courbe de niveau est une courbe décrivant une coupe du terrain à une certaine altitude. Les courbes de niveaux sont séparées de leurs voisines par une dénivellation constante.

Il existe une grande variété de cartes et chacune renferme un certain nombre d'informations à propos de la zone géographique représentée. On y trouve par exemple les forêts, les zones urbaines, les bâtiments, les routes et les rivières etc. Aussi, des modèles de terrains évolués pourraient indexer toutes ces informations. Cependant nous avons, pour l'instant, limité notre étude à la seule extraction des courbes de niveaux et à leur traitement.

Notre but est donc de présenter une chaîne de traitements pour générer un MNT à partir d'une carte topographique numérisée.

**Contexte.** Cette étude s'inscrit dans le projet IPARLA au sein d'INRIA-Futurs<sup>1</sup>. Le projet s'intéresse à la visualisation et à la manipulation de données complexes sur terminaux mobiles communicants (assistants personnels ou PDA, téléphones mobiles). Le but est de développer une chaîne graphique, entre autres, basée sur une architecture

---

<sup>1</sup><http://iparla.labri.fr>

client-serveur adaptative, des modélisations formelles, des descriptions multi-résolution, des visualisations basées pixel et des rendus non photo-réalistes.

La création de MNT constitue une des premières étapes d'une application potentielle de cette chaîne permettant d'offrir un repérage augmenté dans un environnement existant.

## 2 Processus général

L'extraction d'informations sur des cartes topographiques numérisées a engendrée depuis une vingtaine d'années de nombreuses publications que nous allons détailler plus tard.

Nous allons présenter dans l'ordre chronologique le processus que nous avons mis au point. Nous distinguons quatre grandes parties (cf. figure 1) :

1. extraction des courbes de niveaux de la carte,
2. reconstruction automatique ou semi-automatique des courbes altérées,
3. affectation semi-automatique des altitudes des courbes,
4. interpolation des courbes.

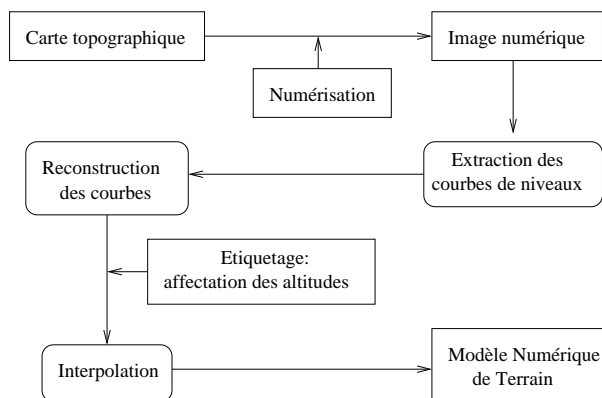


FIG. 1 – Processus de création d'un MNT à partir d'une carte numérisée

Notre étude s'est basée principalement sur [Arr98] et [AS99] qui exposent un processus de création de MNT à partir de cartes topographiques basé sur la morphologie mathématique, de la reconstruction à l'interpolation.

## 3 Extraction des courbes de niveaux

Le but de cette première étape est d'obtenir un masque binaire correspondant aux courbes de niveaux. Ce processus consiste en l'application de plusieurs algorithmes de traitement d'image (filtrage, segmentation couleur, seuillage) afin d'isoler les courbes de niveaux des autres informations.

Nous nous intéressons à des images couleurs RGB 24 bits issues de la numérisation de cartes topographiques (cartes *IGN*, topo-guides, etc.). Elles contiennent, outre les courbes de niveaux, de nombreuses informations telles que les routes, les cours d'eau, les bâtiments, les forêts etc. Chaque type d'information est caractérisé par une couleur spécifique et appartient à une couche (*layer*) d'information particulière. Lors de l'impression de la carte, les différentes couches sont superposées dans un certain ordre.

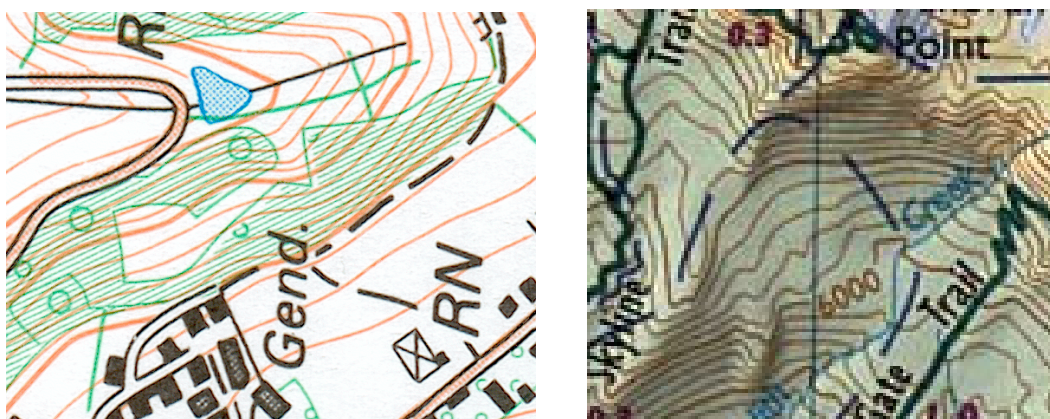
L'étape de la numérisation est assez cruciale pour la suite du traitement. L'ensemble du processus que nous allons décrire reposant sur le traitement des courbes de niveaux de la carte, il est capital que celles-ci répondent aux critères suivants sur l'image obtenue par numérisation :

- les courbes de niveaux doivent être suffisamment épaisses,
- elles doivent être le plus isolées possible les unes des autres,

- la couleur qui les caractérise doit être la plus uniforme possible

Ces réglages doivent donc être effectués en fonction du scanner utilisé, de la qualité de la carte, de son type (topographique seulement, mixte...) ainsi que de son échelle.

Une fois la carte numérisée, il s'agit d'extraire les courbes de niveaux de celle-ci. On utilise pour cela la principale information disponible, c'est à dire la couleur caractéristique des courbes. Par convention, celles-ci sont généralement tracées en marron/orange (cf. figure 2(a)). Cependant, de part la difficulté de réaliser une numérisation parfaite, ou à cause de la complexité de la carte, la phase d'extraction peut être extrêmement délicate à paramétrer, voir quasi impossible à réaliser (cf. figure 2(b))



(a) Extrait d'une carte à l'échelle 1 :25.000 numérisée à 500dpi

(b) Difficulté d'isoler les courbes

FIG. 2 – Extraits de cartes numérisées

De plus, comme l'illustre la figure 2(a) les cartes sont imprimées par superposition des couches de cyan, magenta et jaune. Le moindre décalage entre ces couches ou une infime variation de la quantité d'encre peut entraîner l'apparition de lignes oranges qui ne sont pas des courbes de niveaux. Sur la figure 2(a), on constate également qu'il est difficile de distinguer par leur couleur les morceaux courbes de niveaux qui intersectent d'autres symboles, en l'occurrence les lignes d'une zone hachurée en vert.

Pour réaliser l'extraction des courbes nous effectuons une segmentation couleurpixels de l'image par seuillage. On ne conserve que les pixels dont la couleur appartient à un intervalle fixé correspondant à la plage de couleur des courbes de niveaux. Cependant le modèle de couleur utilisé pour effectuer la discrimination est essentiel. Par exemple, [ASL90] se base sur la teinte (*hue channel*) de la couleur pour discriminer des types de sols sur des cartes pédologiques. [ELA94] transforme l'image RGB dans l'espace de couleur CIE  $L^*u^*v^*$  (*International Illumination Commission*, 1976) pour prendre en compte la chromaticité. [FRL<sup>+</sup>98] calcule une palette de 256 couleurs optimale pour l'image 24 bits. Il discrimine ensuite les pixels du fond par un seuil sur la luminance. Il renforce ensuite la couleur des courbes en affectant à chaque pixel la couleur de son voisin le plus sombre. Enfin, il effectue un regroupement (*clustering*) des couleurs de la palette en se basant sur leur distance dans le modèle HSV. Pour des cartes peu complexes et dont le fond est uniforme, nous utilisons, comme le propose [SG02], des seuils empiriques dans l'espace de couleur HSV pour isoler les courbes.

## 4 Reconstruction

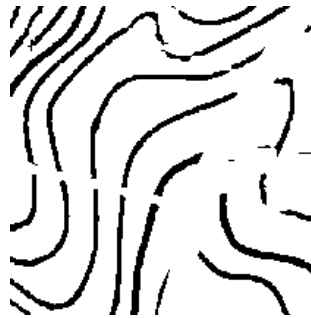
La phase de reconstruction (ou reconnexion) consiste dans un premier temps à affiner (ou squelettiser) les courbes de niveaux, c'est à dire de leur donner une épaisseur de 1 pixel afin de pouvoir les traiter plus facilement. Dans un deuxième temps, on cherche à reconnector les morceaux de courbes de niveaux qui ont été coupées lors de la phase d'extraction. Cette reconnexion est nécessaire pour faciliter la phase d'affectation des altitudes et pour ne pas perdre de la précision lors de l'interpolation du terrain.

## 4.1 Squelettisation

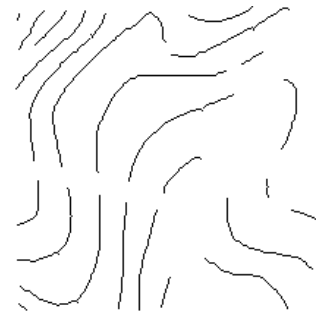
La squelettisation est un problème classique en analyse d'image. Il existe différentes familles d'algorithmes pour résoudre ce problème : simulation de propagation de feu de prairie [Xia89], diagramme de Voronoï [Att95] ou morphologie mathématique [Ser82, GW92]. Nous utilisons cette dernière pour sa facilité de mise en œuvre et ses bons résultats dans le cadre de notre étude. La squelettisation est réalisée par des transformations morphologiques simples. Elle est alors vue comme un processus d'érosion des objets qui en préserve la connexité.



(a) Extrait de carte numérisée



(b) Extraction des courbes



(c) Après squelettisation

FIG. 3 – Affinage

La figure 3(c) montre le résultat obtenu à partir de l'image 3(b). Un des problèmes qui apparaît est la perte d'informations au niveau de l'extrémité des courbes, le processus de squelettisation ayant comme effet de les rogner. [Arr98] propose de détecter le point représentant l'extrémité de la courbe avant d'appliquer une méthode de squelettisation par points d'ancrages.

## 4.2 Reconnexion des lignes interrompues

Une fois les courbes affinées, il s'agit de reconnecter les différents segments de courbes qui se sont retrouvés isolés. La stratégie adoptée consiste à calculer pour chaque extrémité une distance pondérée entre celle-ci et chacune des autres extrémités. On ne considère que les extrémités dans un certain voisinage et qui sont visibles au sens d'un lancer de rayon 2D. Après plusieurs tests on peut estimer une distance raisonnable maximale entre 2 extrémités de courbe à une centaine de pixels. On reconnecte ensuite les extrémités par ordre croissant selon les distances calculées.

**Distances pondérées.** Pour calculer la distance entre deux extrémités nous utilisons deux critères :

- d'une part la distance euclidienne entre celles-ci,
- et d'autre part les angles entre les tangentes discrètes aux deux extrémités et le vecteur formé par les deux extrémités.

Pour reconnecter deux courbes, on cherche le couple d'extrémités qui minimisent ces 2 critères.

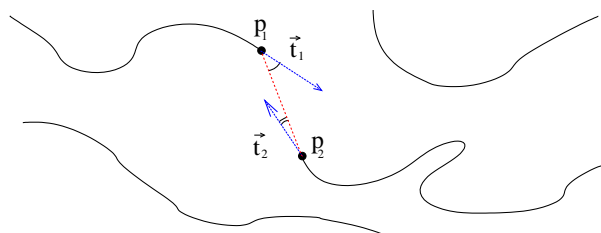


FIG. 4 – Critères de calcul des distances pondérées

Soient  $p_1$  et  $p_2$  deux extrémités, et  $\vec{t}_1$  et  $\vec{t}_2$  deux vecteurs tangents des courbes en ces extrémités (cf. figure 4), nous proposons de calculer la distance pondérée ainsi :

$$d_{p_1 p_2} = \|\vec{p_1 p_2}\| \cdot \left( \mu + \left( 1 + \frac{\alpha_{\vec{t}_1, \vec{p_1 p_2}} + \alpha_{\vec{t}_2, \vec{p_2 p_1}}}{2\pi} \right)^n \right) \quad (4.1)$$

où  $\|\vec{p_1 p_2}\|$  est la norme euclidienne  $\alpha_{\vec{v}_1, \vec{v}_2}$  est l'angle orienté entre les deux vecteurs  $\vec{v}_1$  et  $\vec{v}_2$ , et  $\mu$  et  $n$  sont des coefficients d'autant plus grand qu'on veut pénaliser les angles grands au profit de la distance euclidienne. Après une série de tests, les paramètres  $\mu = 9$  et  $n = 9$  ont donné de bons résultats.

#### Algorithme de reconnexion.

1. Ajouter toutes les extrémités dans une liste  $L$
2. Pour chaque extrémité de  $L$ , ajouter dans une liste  $D$  les distances pondérées calculées selon l'équation 4.1 entre cette extrémité et les autres extrémités de  $L$  qui sont dans un voisinage (distance euclidienne) et que l'on peut atteindre par un lancé de rayon 2D.
3. Trier la liste  $D$  dans l'ordre croissant.
4. Parcourir la liste  $D$  en reliant par une courbe *spline* les extrémités n'ayant pas déjà été préalablement reliées et dont le tracé n'entraîne pas de croisement avec une courbe existante.

La figure 5 illustre le résultat obtenu à partir de la figure 3(c).

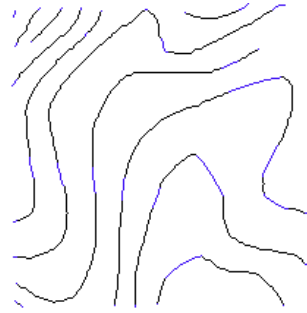


FIG. 5 – Reconnexion des courbes

## 5 Etiquetage

Une fois les courbes de niveaux reconstruites, il faut leur affecter une altitude. Cette tâche peut difficilement être totalement automatisée et doit donc être effectuée en partie manuellement par l'opérateur. Nous avons élaborés des techniques d'affectation semi-automatiques de l'altitude afin de minimiser le nombre d'intervention de l'opérateur. Ces techniques et outils ont été implémentés dans le logiciel AUTOMNT[Pou03] que nous avons mis au point.

## 6 Interpolation

L'interpolation des courbes de niveaux a donné lieu à de nombreux travaux [GF98]. On distingue deux familles de techniques :

- les algorithmes générant des réseaux de triangles irréguliers (*Triangulated Irregluar Network, TIN* [Fra96]), que nous présentons succinctement dans la section 6.1.
- les algorithmes traitant le terrain comme une grille où chaque point dont l'altitude n'est pas connue est interpolé, qui sont étudiés dans la section suivante. Nous présenterons plus en détail une méthode basée sur les fonctions de bases radiales que nous avons mise au point.

## 6.1 Les Réseaux de Triangles Irréguliers

L'interpolation par réseaux de triangles irréguliers, dont la première approche date de 1973 par Franklin [Fra96], consiste à générer un modèle de terrain triangulé. Le réseau est créé en échantillonnant des points appartenant aux courbes de niveaux et en les reliant judicieusement trois à trois pour former des triangles, généralement par triangulation de Delaunay. Plus récemment, [HSS03] propose une technique d'interpolation des courbes de niveaux C1-continue élégante, et permettant notamment de bien modéliser les crêtes et lignes de selle. L'avantage de ces techniques est qu'elles engendrent directement un modèle 3D. Cependant la triangulation obtenue ne permet pas d'appliquer directement des techniques efficaces de visualisation par de niveau de détail telles que le *roaming* [DWa97].

## 6.2 Le terrain vu comme une grille

### 6.2.1 Interpolation par rayonnement

Jones, Hamilton et Johnson [JHJ86] proposent une technique d'interpolation pour les courbes de niveaux qui consiste à envoyer des rayons dans 8 directions à partir des points dont on cherche l'altitude jusqu'à ce qu'ils intersectent des courbes de niveaux. L'altitude finale est ensuite calculée par une moyenne pondérée en fonction de l'inverse de la distance. Cette technique est peu complexe mais ne donne pas des résultats suffisamment précis et réalistes comme le montre la figure 6(c). Dans le cadre d'un logiciel de SIG, elle permet cependant d'obtenir instantanément l'estimation de l'altitude d'un point.

### 6.2.2 Interpolation par Equations aux Dérivées Partielles

L'interpolation grâce aux EDP (*PDE*), issue d'un modèle physique, est la plus utilisée pour interpoler une grille d'altitudes [Fra00, Fra96]. [Tob79] propose une introduction à l'utilisation des EDP en cartographie.

Une équation de Laplace est une EDP satisfaite par la conduction de la chaleur dans le milieu homogène d'une plaque de métal mince (*thin plate*) : quand on chauffe certains points à une température fixée, les autres points de la plaque montent progressivement en température jusqu'à atteindre une température qui correspond à la température moyenne par rapport aux températures de leurs voisins immédiats. Pour l'interpolation de courbes de niveaux, on utilise ce même principe en remplaçant la chaleur par l'altitude.

L'équation à résoudre est la suivante :

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = 0 \quad (6.1)$$

Cela revient à résoudre les équations suivantes sur une grille de points  $(i, j)$  :

$$4z_{i,j} = z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} \quad (6.2)$$

Concrètement, cela signifie que l'altitude de chaque point doit être la moyenne de ses 4 voisins. Le problème de cette technique est la complexité de sa résolution. Sur une grille de  $1000 \times 1000$ , il y a jusqu'à plus  $N = 1000^2 = 1M$  d'inconnues. Pour résoudre le problème on utilise donc des méthodes itératives qui permettent d'obtenir des solutions approchées. Une solution est d'utiliser une technique multi-grille qui consiste à résoudre l'EDP sur des grilles grossières, puis à raffiner la solution sur des grilles plus fines. De cette manière, on peut réaliser l'interpolation sur des grilles contenant des millions de points.

Cependant, la surface obtenue a tendance à s'affaisser entre les courbes de niveaux ce qui peut poser un problème de crédibilité du terrain. [Fra00] propose d'initialiser le processus avec une estimation obtenue par la méthode des moindres carrés. [Chi03], expose les résultats obtenus grâce à cette méthode.

### 6.2.3 Fonctions de Bases Radiales et Partition de l'Unité

Il y a plus de vingt ans, Franke [Fra82] montrait que les fonctions de bases radiales (*RBF*) étaient une méthode stable pour résoudre le problème de l'interpolation de données dispersées. Ces dernières années, de nombreux travaux ont étudié les RBF pour interpoler des surfaces implicites. Celles-ci sont définies par des nuages de points

généralement obtenus par l'intermédiaire d'un scanner-3D. Le problème de cette technique se situe sur le plan de la complexité. En effet, le calcul de la fonction d'interpolation par RBF nécessite de résoudre un système linéaire. Or les RBF ont un support global et les équations mènent à un système linéaire extrêmement dense, limitant ainsi la reconstruction à quelques centaines de points. Pour palier à cela, différentes approches et techniques ont été proposées, et notamment des méthodes de partition de l'unité que nous présenterons plus loin.

Dans le cadre de l'interpolation de terrain, la démarche consiste à considérer notre MNT comme une surface implicite qu'il faut reconstruire à partir des points des courbes de niveaux. Nous nous sommes basés sur [TRS03, Reu03] qui présentent une méthode de reconstruction de surfaces implicites à partir de grands ensembles de points dispersés grâce aux RBF et à la partition de l'unité.

**Définitions.** Soit  $P = \{p_1, \dots, p_N\}$  un ensemble de points de l'espace euclidien  $\mathbb{R}^k$ , et un ensemble de valeurs  $\{h_1, \dots, h_N\}$ , on cherche une fonction  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  telle que :

$$f(p_i) = h_i \quad (6.3)$$

Pour obtenir une reconstruction par RBF d'un ensemble de points  $P$ , il faut trouver la fonction  $f$  définie en un point  $p$  comme la combinaison linéaire de fonctions de base :

$$f(p) = \pi(p) + \sum_{i=1}^N \lambda_i \phi(\|p - p_i\|) \quad (6.4)$$

où  $\pi$  un polynôme  $\pi(p) = \pi_0 + \sum_{i=1}^k \pi_i p^{(i)}$  où  $p^{(i)}$  est le  $i$ -ème composant du vecteur  $p$ ,  $\lambda_i$  des poids,  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  une fonction de base et  $\|p_i - p_j\|$  est la distance euclidienne.

La fonction de base  $\phi$  est une fonction de  $r$  correspondant à une distance radiale. On choisit généralement  $\phi$  parmi celles-ci :  $\phi(r) = r$ ,  $\phi(r) = r^3$ ,  $\phi(r) = r^2 \log(r)$  ou  $\phi(r) = \sqrt{r^2 + c^2}$ .

Les fonctions de bases radiales sont utilisées dans des problèmes d'interpolation, d'approximation, de reconstruction à partir de données non ordonnées et non uniformément espacées. En fonction du choix de  $\phi$ , la fonction  $f$  minimise certains paramètres (l'énergie, la courbure).

**Résolution.** Pour déterminer l'équation 6.4, il faut calculer les poids  $\omega_i$  et les coefficients du polynôme  $\pi$ . Cela revient à résoudre un système linéaire défini par  $N$  équations du type de l'équation 6.4. Ce système est sous-déterminé car il possède  $N$  contraintes et  $N + (k + 1)$  inconnues. On introduit donc les  $k + 1$  contraintes supplémentaires suivantes :

$$\sum_j \lambda_j = \sum_j \lambda_j \pi_1 = \dots = \sum_j \lambda_j \pi_k = 0 \quad (6.5)$$

Les équations 6.3, 6.4 et 6.5 mènent à un système linéaire de type  $Ax = b$  avec :

$$\begin{aligned} A &= \begin{pmatrix} \Phi_{1,1} & \dots & \Phi_{1,N} & p_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \Phi_{N,1} & \dots & \Phi_{N,N} & p_N & 1 \\ p_1 & \dots & p_N & 0 & 0 \\ 1 & \dots & 1 & 0 & 0 \end{pmatrix} \\ x &= (\omega_1, \dots, \omega_N, \pi_1, \dots, \pi_k, \pi_0)^T \\ b &= (h_1, \dots, h_N, 0, \dots, 0) \end{aligned} \quad (6.6)$$

avec  $\Phi_{i,j} = \phi(\|p_i - p_j\|)$ .

Soit  $N$  le nombre de contraintes de  $f$  (ie. de points connus), et  $M$  le nombre de points dont on cherche la valeur par évaluation de  $f$ , la complexité de résolution du système 6.6 par décomposition LU est en  $O(N^3)$  et la complexité d'évaluation de la fonction  $f$  de l'équation 6.4 est en  $O(MN)$ . Ceci rend difficile, voire impossible l'application pratique de cette méthode pour les problèmes dont le nombre de contraintes est grand et en particulier l'interpolation d'un MNT à partir de courbes de niveaux.



**Partitionnement hiérarchique de l'unité.** L'idée de l'approche par partition de l'unité est de diviser le domaine global d'intérêt en plusieurs sous-domaines où le problème peut être résolu localement. La solution globale du système est ensuite obtenue en rassemblant les solutions locales.

Nous utilisons l'algorithme de partitionnement hiérarchique inspiré de [Reu03]. Le partitionnement se fait de manière récursive et engendre la construction d'un arbre binaire parfaitement équilibré. On divise chaque région  $\Omega$  en 2 sous-régions selon le plus grand axe de son rectangle englobant de manière à avoir dans chaque sous-région le même nombre de contraintes avec un facteur de chevauchement de l'ordre de quelques pour cents. On arrête la subdivision lorsque le nombre de points est inférieur à une borne  $T_{max}$ . On calcule ensuite dans chaque feuille de l'arbre la fonction RBF associée.

Dans la phase d'évaluation, pour chaque point  $x$ , on évalue les fonctions RBF des feuilles auxquelles le point appartient puis on rassemble les solutions de manière ascendante, en appliquant à chaque nœud une moyenne pondérée fonction de la distance du point par rapport au centre des rectangles englobants. La solution globale  $f(x)$  est obtenue à la racine de l'arbre.

[Reu03] établit que la complexité pour la création de l'arbre est en  $O(N \log N)$ . Dans les feuilles, le calcul de la fonction RBF locale est en  $O(1)$  vu que le nombre de contraintes est borné par la constante  $T_{max}$ . Le nombre de feuilles étant proportionnel à  $N$ , toutes les fonctions peuvent être déterminées en  $O(N)$ . L'évaluation totale de la fonction globale  $f$  se fait en temps  $O(M \log N)$ .

La grille de la figure 6(a) fait  $512 \times 512 = 262.144$  points dont 3.154 appartiennent à des courbes de niveaux. Le tableau suivant présente le nombre de régions créées par partitionnement ainsi que le temps de calcul nécessaire<sup>2</sup> à l'interpolation complète de la grille :

$T_{max}$	Nombre de régions (feuilles)	Temps de calcul
300	128	27s
400	128	29s
600	64	62s
1000	32	138s

Nous avons constaté que les valeurs  $T_{max} \geq 600$  donnaient des résultats satisfaisants. Avec des valeurs plus petites le risque d'avoir des régions qui n'ont qu'une seule courbe d'altitude comme contrainte est plus grand. La conséquence est alors la présence de zones anormalement plates ou de zones de plissements aux frontières des régions. La figure 6(d) montre le résultat de l'interpolation obtenu. De manière générale, nous avons constaté que le résultat est plus lisse que celui obtenu avec les techniques classiques par rayonnement ou Laplacien. On ne constate pas d'effet d'affaissement entre les courbes, et les terrains obtenus sont visuellement satisfaisants. On peut toutefois reprocher à ce modèle d'engendrer des terrains trop lisses, et peu réalistes pour des zones très montagneuses.

## 7 Conclusion et perspectives

Nous avons présenté un processus de création d'un modèle numérique de terrain à partir d'une carte numérisée. Reste cependant à approfondir la phase la plus importante, l'extraction de courbes de niveaux pour parvenir à exploiter des cartes complexes. Notre étude s'est portée sur la seule extraction des courbes de niveaux. Mais l'extraction des autres informations de la carte permettrait de générer un modèle plus complet. Nous envisageons d'extraire, par exemple, les routes, les bâtiments, les forêts et les cours d'eau afin de pouvoir les intégrer au MNT.

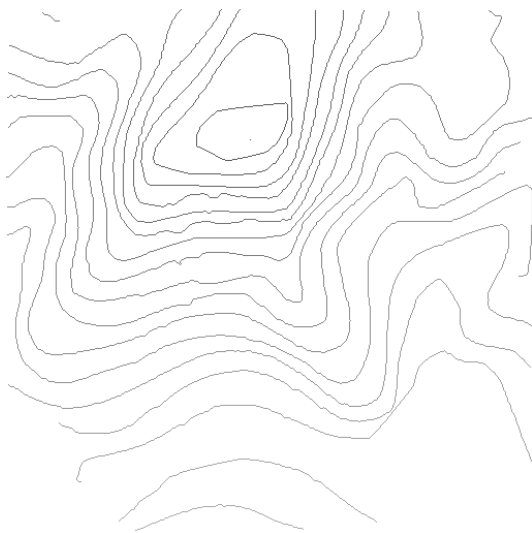
**Visualisation et navigation sur PDA.** Comme nous l'avons signalé dans l'introduction, ce travail s'inscrit dans un projet destiné à l'implémentation d'une chaîne de traitements sur terminaux mobiles. Les caractéristiques de ces machines sont les suivantes : écrans de petite taille et de faible résolution (généralement aux alentours de  $480 \times 320$  pixels) et sans processeur dédié à l'accélération du rendu 3D (signalons cependant que différents constructeurs y travaillent actuellement), mémoire de masse relativement limitée (quelques centaines de mégaoctets) et capacité de connexion à un réseau informatique local ou global, en particulier à un système de localisation par satellite (GPS) afin offrir à l'utilisateur un repérage de sa position dans l'espace. On imagine cependant que l'écart va se creuser dans les prochaines années entre les performances d'une station de travail et ces machines. La principale limitation

<sup>2</sup>Tests effectués sur un AthlonXP 2GHz avec 1Go de RAM

restera celle concernant la taille de l'écran. Il est donc nécessaire d'implémenter des techniques de visualisation et de navigation des terrains qui prennent en compte ces caractéristiques.

Pour la visualisation, notre investigation porte sur des rendus 3D par polygones que l'on peut générer notamment grâce à des ports de la bibliothèque OpenGL (OpenGL/ES ou PocketGL). Un rendu par points pourrait également s'avérer une bonne technique. Le but n'étant pas forcément de créer un environnement virtuel photo-réaliste, nous nous intéressons également à des techniques de rendus non photo-réalistes.

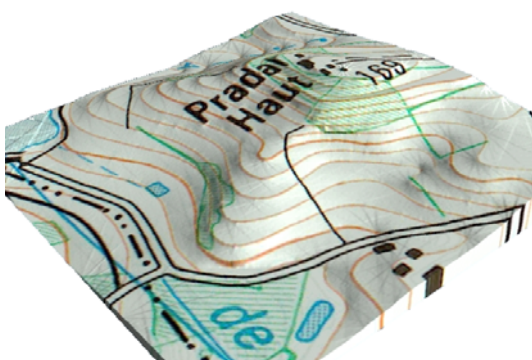
Enfin, la taille d'un MNT pouvant être énorme, il sera nécessaire d'étudier des méthodes permettant d'optimiser au mieux le rendu pour permettre une navigation fluide. Des techniques de rendu par niveaux de détail devront être implémentées qui consiste à adapter le nombre de facettes du modèle 3D en fonction de la distance du point de vue et de la rugosité du terrain). Nous étudierons également un modèle client-serveur permettant de transférer tout ou partie du MNT au terminal à partir d'un serveur distant.



(a) Courbes de niveaux



(b) Interpolation par RBF avec partitionnement hiérarchique



(c) Vue 3D d'une interpolation par rayonnement



(d) Vue 3D de l'interpolation par RBF et partitionnement de l'unité

FIG. 6 – Exemple d'interpolation

## Références

- [Arr98] P. Arrighi. Reconstruction 3d de cartes numérisées. Master's thesis, Université Montpellier II, 1998.
- [AS99] P. Arrighi and P. Soille. From scanned topographic maps to digital elevation models. In *Proceedings of Geovision '99*, 1999.
- [ASL90] M. Ansault, P. Soille, and J. Loodts. Mathematical morphology : a tool for automated gis data acquisition from scanned thematic maps. *Photogrammetric Engineering and Remote Sensing*, 56(9) :1263–1271, 1990.
- [Att95] D. Attali. *Squelette et Graphes de Voronoï 2D et 3D*. PhD thesis, Université Joseph Fourier, 1995.
- [Chi03] J. Childs. Development of a two-level iterative computational method for solution of the franklin approximation algorithm for the interpolation of large contour line data sets. Master's thesis, Rensselaer Polytechnic Institute, Troy, NY 12180, may 2003.
- [DWa97] M. A. Duchaineau, M. Wolinsky, and al. Roaming terrain : a real time optimally adaptating meshes. *IEEE Vizualisation 97*, pages 81–88, 1997.
- [ELA94] N. Ebi, B. Lauterbach, and W. Anheier. An analysis system for automatic data acquisition from colored scanned maps. *Machine Vision and Applications*, 7(3) :148–164, 1994.
- [Fra82] R. Franke. Scattered data interpolation : Tests of some methods. *Mathematics of Computation*, 38(157) :181–200, 1982.
- [Fra96] W.R. Franklin. Elevation data operations. *Dagstuhl - Workshop on computational cartography*, 1996.
- [Fra00] W. R. Franklin. Applications of analytical cartography. *Cartography and geographic information system*, april 2000.
- [FRL<sup>+</sup>98] X. Fernandez, F. Riveiro, O.M. Lopez, X. Rial, J.V. Oliveira, and F. Martin. A vectorizer for color topographic maps. In *Proceedings of the IASTED International Conference - Signal and Image Processing*, October 1998.
- [GF98] M.B. Gousie and W.R. Franklin. Converting elevation contours to a grid. *Eighth International Symposium on Spatial Data Handling (SDH)*, 1998.
- [GW92] R. Gonzalez and R. Woods. *Digital Image Processing*, pages 518–548. Addison-Wesley Publishing Company, 1992.
- [HSS03] K. Hormann, S. Spinello, and P. Schröder. C1-continuous terrain reconstruction from sparse contours. In *To appear in Visual Modelling and Visualization*, Munich, 2003.
- [JHJ86] T. A. Jones, D.E. Hamilton, and C.R. Johnson. *Contouring geologic surfaces with the computer*. Van Norstand Reinhold Company Inc., New York, 1986.
- [MGL91] I. Moore, R. Grayson, and A. Ladson. Digital terrain modelling : a review of hydrological, geomorphological, and biological applications. *Hydrological Processe*, 5 :3-30, 1991.
- [Pou03] J. Pouderoux. Création automatique d'un modèle numérique de terrain. Master's thesis, LaBRI, Université Bordeaux I, juin 2003.
- [Reu03] P. Reuter. *Reconstruction and rendering of implicit surfaces from large unorganized point sets*. PhD thesis, LaBRI, Université Bordeaux I, 2003.
- [Ser82] J. Serra. *Image analysis and mathematical morphology*. Academic Press, 1982.
- [SG02] S. Spinello and G. Greiner. Automatic contour line recognition from scanned topographic maps. In *Poster section*, Saint-Malo, France, july 2002. Fifth International Conference on Curves and Surfaces.
- [Tob79] W. Tobler. Smooth pycnophylatic interpolation for geographic regions. *J. Am. Stat. Assn.*, 74(367) :519–536, 1979.
- [TRS03] I. Tobor, P. Reuter, and C. Schlick. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. Technical Report RR-1301-03, LaBRI, may 2003.
- [Xia89] I. Xia. Skeletonization via the realisation of the fire front's propagation and extinction in digital binary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :1076–1086, 1989.